

DISCLAIMER

The modifications suggested in this guide have worked for infoVia, but no assurances are provided. Individual Wherescape environments can differ and there is potential for these changes to conflict with other customizations, cause environmental oddities, or just not work. Finally, this is a set of modifications to a WhereScape field solution. WhereScape does not support field solutions and will not provide support for modifications to field solutions. This guide was produced in hopes of helping the WhereScape community. *USE AT YOUR OWN RISK*

The Record Tracking Satellite solution by WhereScape provides automated tracking for Hub Business Keys and Link Business Keys when no external solution (e.g. Change Data Capture) is available and full data loads are being performed. The solution generates an additional satellite for each Hub and Link which will perform a comparison of the recorded business keys with the incoming business keys from the current load. This solution is enabled at the data connection level. A video demonstrating the solution can be found here: <https://support.WhereScape.com/support/discussions/topics/4000343053>

The un-edited field solution from WhereScape supports a data environment which is non-indexed, assumes each data data source for link or hub has a corresponding satellite, and that a hub or link has a single source. This guide includes modification suggestions to eliminate those restrictions and has worked on Wherescape environments for both Snowflake and MS SQL Server. Modifications for other RDBMS environments will require adjusting the ANSI SQL to the desired syntax.

Requirements:

- Download the zip file containing the necessary components from [<https://support.WhereScape.com/support/discussions/topics/4000343051>]

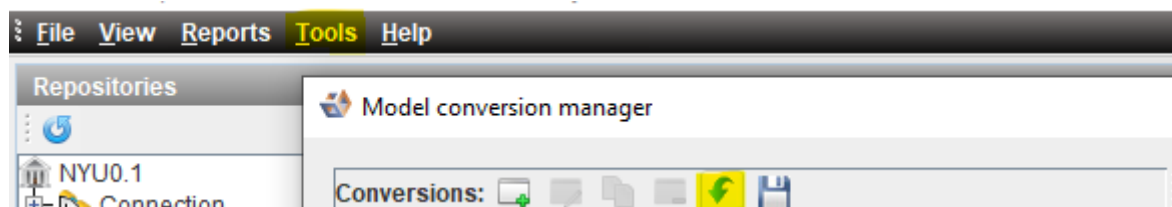
WhereScape 3D Enviroment Changes

Imports

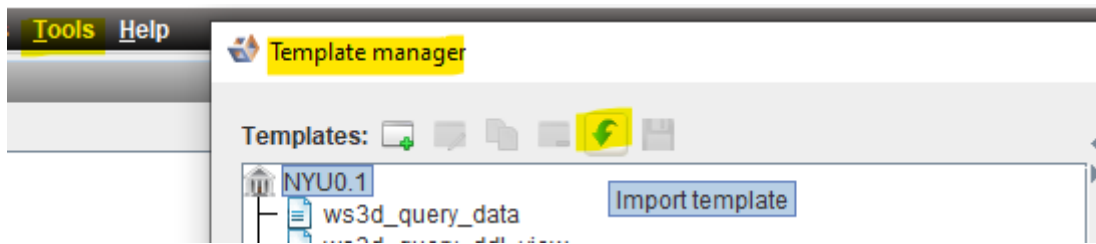
Import the Model Conversion Rule (MCR) file into your WhereScape 3D Environment

Select Tools > Manage Model Covnersions

Use the green arrow to import the downloaded Model Conversion Rules



Import the Template file into your WhereScape 3D Environment



Environment Settings

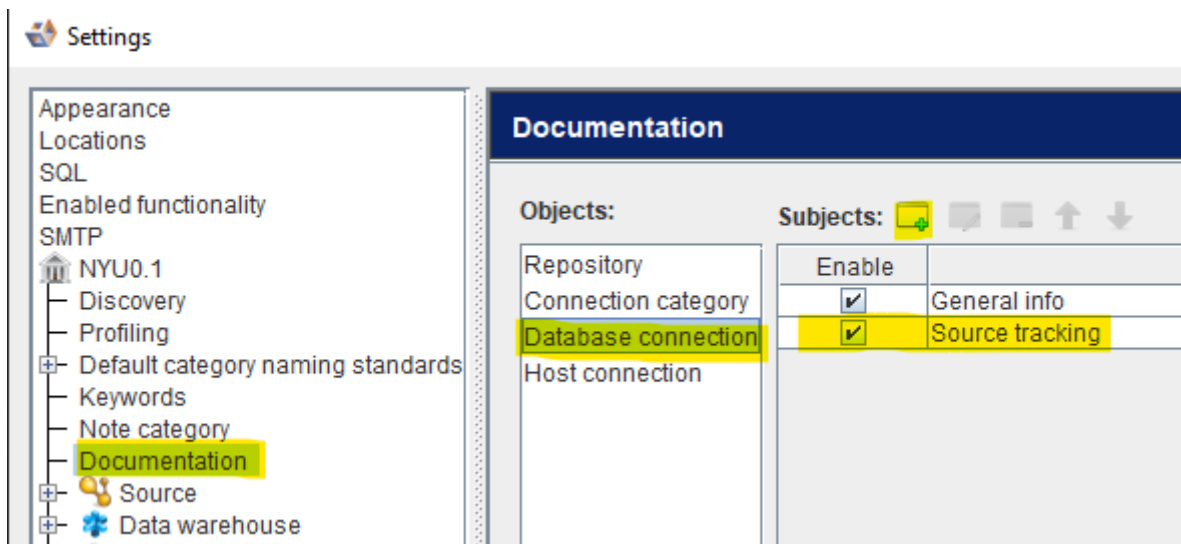
Add user documentation options

Open settings (Tools>Settings)

Navigate to the "Documentation" section in the left window

Select "Database connection" in the Objects list

Select the windows with a green plus and enter the label "Source tracking" with the description "Source tracking" Capitalization *does* matter. Click OK



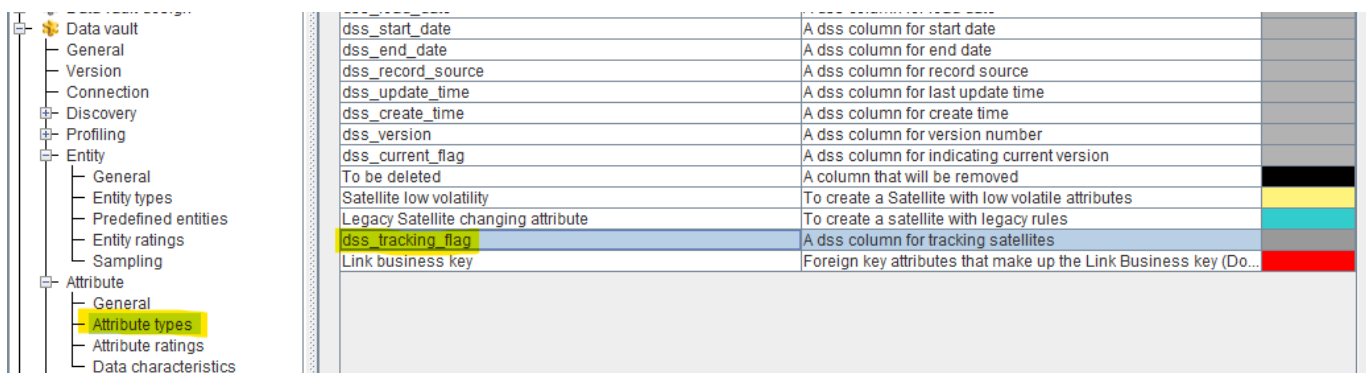
Add tracking Attribute Type

Open settings (Tools>Settings)

Navigate to Data Vault>Attribute>Attribute Types

Click the window with the green plus and enter "dss_tracking_flag" for the name and a "A dss column for indicating record tracking" for the description. Make the color Grey

Add the same attribute types to the Load and Staging model



Alter "User Documentation" for connections

Right click on a connection and select "Enter User Documentation"

Navigate to the "Source tracking" tab and enter the letter "Y" . Capitalization *does* matter

Click OK

Repeat for any other connections to enable Source Tracking

Adjust Model Conversion Rules & Template

Adjust ws3d_dv_record_tracking template

Replace the "Generate the query" portion of the script with the below. Doing so makes the following changes:

- Adjusts the join condition from hub/link to stage table to only join on the hash key instead of the hash key and dss_source_record
- Table specific aliases are replaced with a standard alias to reduce multi-source issues
- Satellites are no longer queried for hash keys

```
{#- ===== -#}
{#- Generate the query -#}
{#- ===== -#}
{%- set graphs = createSourceGraphBySet(table, sourceSet, false, false) -%}

{%- if graphs | length > 0 -%}
{%- get selectCols from table.columns as col where col.attributeTypes contains
"Hub hash key" or col.attributeTypes contains "Link hash key" or
col.attributeTypes contains "dss_record_source"-%}
SELECT 1{%br%}
FROM [TABLEOWNER].[{{dvObject.name}}] {%br%}
{#{%- if sats | length > 0 -%}
INNER JOIN ({%br%}
{%- for sat in sats -%}
    {%- for selectCol in selectCols -%}
        {%- if loop.first -%}SELECT {% else -%}, {% endif -%}{{selectCol.name}}{%
br %}
    {%- endfor -%}
    FROM [TABLEOWNER].[{{ sat }}]{%br%}
    {%- if not loop.last -%}UNION{%br%}{%- endif -%}
{%- endfor -%}
) tgt {%br%}
{%- from table.columns as col where col.attributeTypes contains "Hub hash key" or
col.attributeTypes contains "Link hash key" -%}
    {%- if loop.first -%}ON {% else -%}AND {% endif -%}
    {{dvObject.name}}.{{col.name}} = tgt.{{col.name}}{%br%}
{%- endfrom -%}
{%- else %}#}tgt {%br%}
{#{%- endif %}#}
FULL OUTER JOIN [TABLEOWNER].[{{stage.name}}] stage_table{%br%}
{%- for col in selectCols -%}
    {% if col.attributeTypes contains "dss_record_source" %}
    {% else %}
```

```
{%- if loop.first -%}ON {% else -%}AND {% endif -%}
stage_table.{{col.name}} = tgt.{{col.name}}{%br%}
CROSS JOIN (SELECT TOP 1 dss_record_source FROM [TABLEOWNER].[{{stage.name}}])
source_table{%br%}
{% endif %}
{%- endfor -%}

{%- endif -%}
```

Adjust field_rvls - Record tracking satellite staging model conversion rule

Make a copy of *field_rvls - record tracking satellite staging* model conversion rules and then adjust the created copy by replacing the code for each rule's listed field. These changes help prevent errors generating from hub or links having multiple sources.

Rule: Add Hash Key transformation

- Field - Match Source set

```
{{- sourceSet.name -}}
```

-
- Field - Change

```
{%- for src in transformationColumn.sourceColumns -%}
  {%- if loop.first -%}
    {%- from src.table.entityTypes as eType where eType.name == "Stage-Data Vault
Stage" -%}
      COALESCE(tgt.{{transformationColumn.name}}, stage_table.
{{transformationColumn.name}})
    {%- endfrom -%}
  {%- endif -%}
{%- endfor -%}
```

Rule: Add Record Source

- Field - Match Source set

```
{{- sourceSet.name -}}
```

- Field - Change

```
source_table.dss_record_source
```

Rule: Add Load Date transformation

SNOWFLAKE NOTE: If working in a Snowflake environment replace "@v_current_datetime" with "CURRENT_TIMESTAMP" in the change code below

- Field - Match Source set

```
{{- sourceSet.name -}}
```

- Field - Change

```
{%- for src in transformationColumn.sourceColumns -%}  
  {%- if loop.first -%}  
    {%- from src.table.entityTypes as eType where eType.name == "Stage-Data Vault  
Stage" -%}  
      COALESCE(stage_table.{{transformationColumn.name}}, @v_current_datetime)  
    {%- endfrom -%}  
  {%- endif -%}  
{%- endfor -%}
```

Rule: Add tracking flag transformation

- Field - Match Source set

```
{{- sourceSet.name -}}
```

- Field - Change

```
{%- from table.columns as column where column.attributeTypes contains "Hub hash  
key" or column.attributeTypes contains "Link hash key" -%}  
  CASE WHEN stage_table.{{column.sourceColumns[0].name}} IS NULL THEN 0 ELSE 1
```

END

{%- endfrom -%}

Add Rule:

Rule Name - Delete tracking flag from DV Stage

Rule Type - Delete attributes

- Add matching for entity type "Stage-Data Vault Stage" and attribute name "dss_tracking_flag"

The screenshot shows a software interface for configuring rules. At the top, there are tabs for 'Description', 'Rules', and 'Validation'. Below the tabs is a toolbar with icons for adding, editing, deleting, and moving rules, along with a search box labeled 'Type here to filter rules'. A table lists the rules:

Name	Rule type
▼ Delete tracking flag from DV stage	Delete attributes
▼ Delete dss_tracking_flag attributes	Assign attribute type

Below the table, there are radio buttons for 'Match all of the conditions (AND)' (selected) and 'Match any of the conditions (OR)'. Two conditions are listed:

- Has entity type: Stage-Data Vault Stage
- AND
- Has attribute name like: dss_tracking_flag

There is a 'Case sensitive' checkbox which is unchecked. At the bottom, there is a dropdown menu labeled 'Add matching for'.

Add Rule:

Rule Name - Delete dss_tracking_flag_attributes

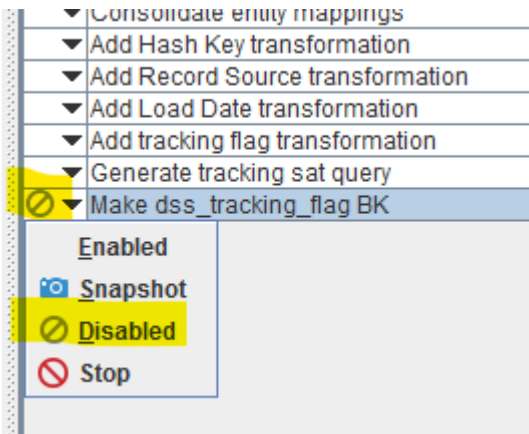
Rule Type - Assign Attribute Type

- Add matching for "as attribute type:" and select dss_tracking_flag from the drop-down
- Check the box for the option "Remove matched attribute types"

Index Warning!

The dss_tracking_flag is assigned the primary key attribute. For SQL Server, and possibly others, this will cause Red to attempt to generate a unique index on that column and it will not be able to enter more than a single

1 and 0 which will stop the update procedure from completing. Disable the "Make dss_tracking_flag BK" rule to prevent this behavior.*



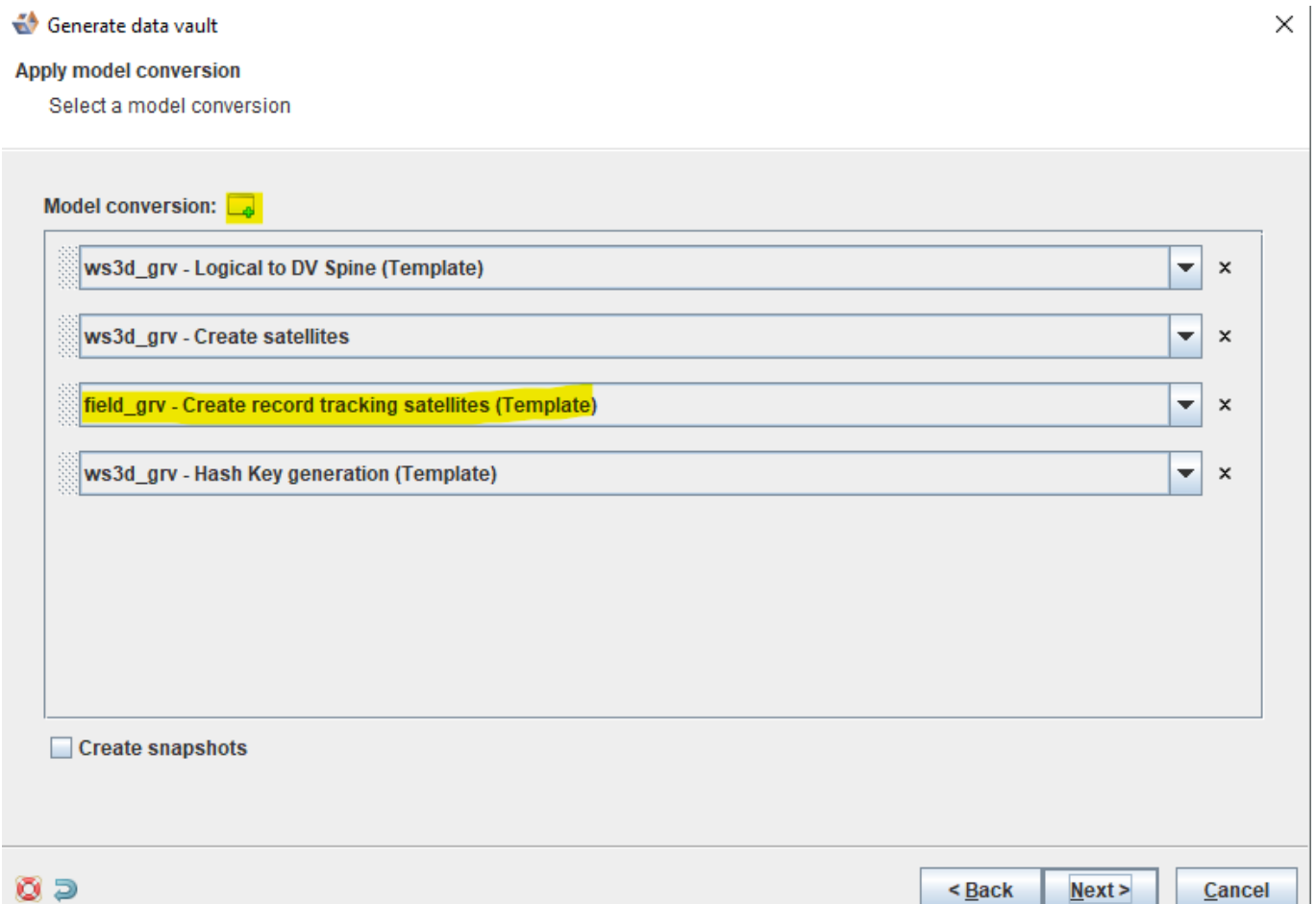
Modeling Wizards

Converting from Data Vault Design to Data Vault

From a Data Vault Design model execute the "Generate data vault" wizard

Proceed through wizard as normal until "Apply model conversion" step is reached

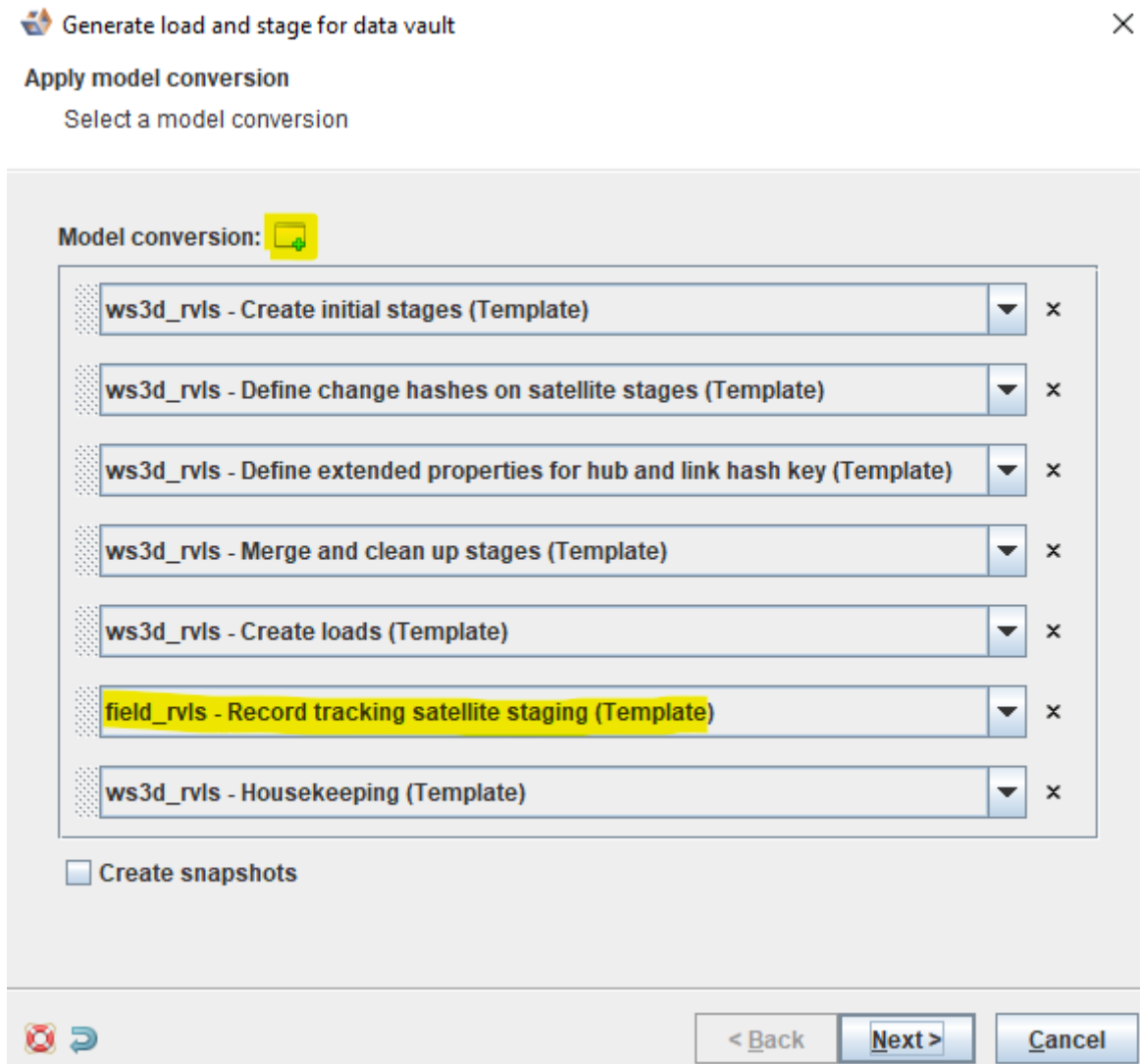
At the "Apply model conversion" step add the rule "field_grv - Create record tracking satellites (Template)" immediately following the standard "ws3d_grv - Create satellites (Template)" rule You will need to add a model conversion rule slot by clicking the window with the green plus and then selecting from the drop-down windows as needed.



Converting from Data Vault to Load and Stage

From a Data Vault model execute the "Generate load and stage" wizard Proceed through wizard as normal until "Apply model conversion" stage is reached

At the "Apply model conversion" add the copy of "field_rvls - Record tracking satellite staging (Template)". This copy should contain the changes listed in the section "Adjust field_rvls - Record tracking satellite staging model conversion rule". It should be added immediately following the standard "ws3d_rvls - Create load (Template)" rule. You will need to add a model conversion rule slot by clicking the window with the green plus and then selecting from the drop-down windows as needed.



WhereScape Red Environment Changes

NOTE: WhereScape recommends using copies of templates to make changes and leaving the originals unaltered. See WhereScape Red help files Data Vaults > Building Data Vault Objects > Changing the Data Vault Hash for an example

Adjust Data Vault Utility Template

Open the utility_dv template appropriate to your environment (e.g. wsl_sqlserver_utility_dv for SQL Server, wsl_snowflake_utility_dv for Snowflake)

Find the code block controlling the where clause for satellites.

Normally the change hash would serve to detect changes. For this solution the dss_tracking_flag column is

used. The changes below detect if an object is a tracking satellite and then checking the `dss_tracking_flag` status. Replace the macro code or duplicate the logic.

```
{# -- This macro inserts the where not exists query for a satellite -- #}
{% macro addSatWhereNotExists(indent="      ") %}
{%- if settings.sourceJoinDetails.where | trim != "" %}
AND NOT EXISTS ({{br%}}
{%- else %}
WHERE NOT EXISTS ({{br%}}
{%- endif %}
{{indent}}SELECT 1{{br%}}
{{indent}}FROM [TABLEOWNER].[{{table.name}}] {{table.name}}{{br%}}
{%- from table.columns as column where column.linkHashKey or column.hubHashKey or
column.changeHashKey or column.dssStartDate or column.keyType.name ==
"MultiActiveNatural" or column.keyType.name == "MultiActiveSequence" or
column.name == "dss_tracking_flag"-%}
  {%- if loop.first %}{{indent}}WHERE {% else %}{{indent}}AND {% endif -%}
  {%- if column.dssStartDate %}current_rows.{{column.name}}
  {%- else -%}
  {{column.source}}{%- endif %} = {{table.name}}.{{column.name}}{{br%}}
{%- endfrom %}
{{indent}})
{%- from table.columns as column where column.name == "dss_tracking_flag"-%}
{{indent}} AND (current_rows.dss_record_source IS NOT NULL OR
stage_table.dss_record_source IS NOT NULL)
{%- endfrom %}
{% endmacro %}
```

Find the code block for the returning the current satellite rows.

Multiple sources on a hub or link need the code below which detects a tracking satellite by the `dss_tracking_flag` column and then includes `dss_record_source` in the group by statement and join so multiple sources can be tracked.

```
{# -- This macro inserts the query to find the current row for a satellite -- #}
{% macro addSatCurrentVersion(indent="      ") %}
LEFT OUTER JOIN ({{br%}}
{{indent}}SELECT
{%- from table.columns as column where column.hubHashKey or column.linkHashKey or
column.keyType.name == "MultiActiveNatural" or column.keyType.name ==
"MultiActiveSequence" or column.name == "dss_tracking_flag" %}
  {%- if loop.first %} {% else %} {{indent}}      , {% endif %}
  {%- if column.name == "dss_tracking_flag" -%} {{table.name}}.dss_record_source
{{br%}} {% else %}
  {{table.name}}.{{column.name}}{{br%}}{%- endif -%}
{%- endfrom %}
```

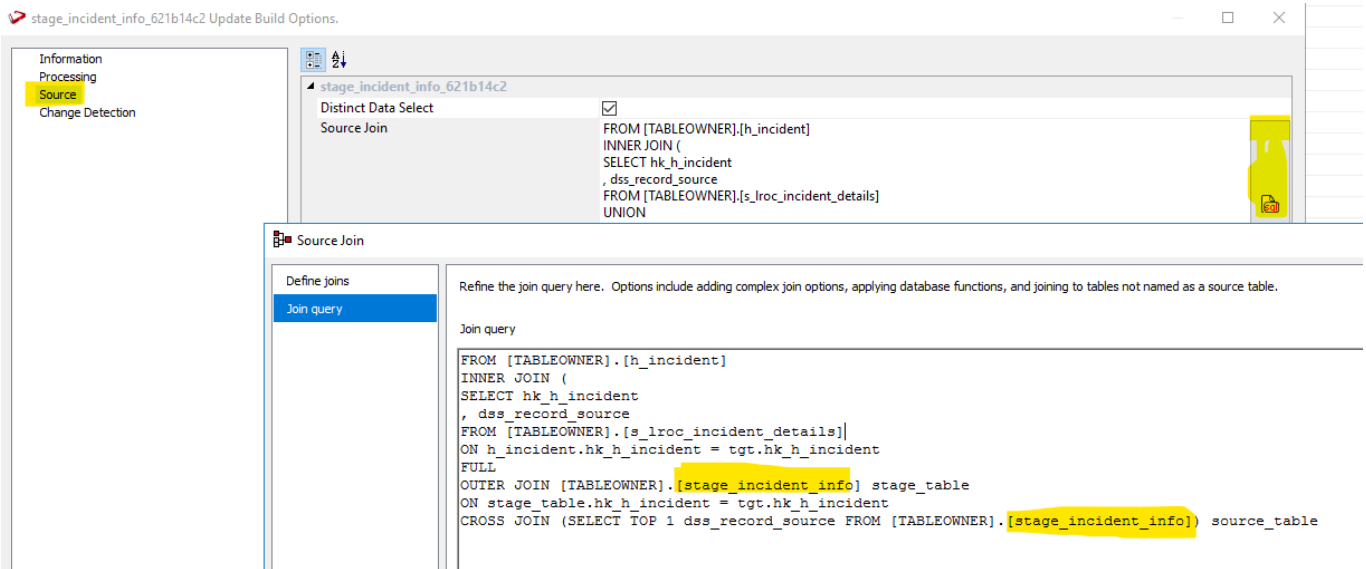
```

{% from table.columns as column where column.dssStartDate or column.dssVersion %}
  {{indent}} , MAX({{table.name}}.{{column.name}}) AS {{column.name}}{%br%}
{% endfrom %}
{{indent}}FROM [TABLEOWNER].[{{table.name}}] {{table.name}}{%br%}
{{indent}}{%br%}
{% from table.columns as column where column.hubHashKey or column.linkHashKey or
column.keyType.name == "MultiActiveNatural" or column.keyType.name ==
"MultiActiveSequence" or column.name == "dss_tracking_flag" %}
  {% if loop.first %}{{indent}}GROUP BY {% else %}{{indent}}, {% endif %}
  {% if column.name == "dss_tracking_flag" -%} {{table.name}}.dss_record_source
{%br%}{% else %}
  {{table.name}}.{{column.name}}{%br%}{%- endif -%}
{% endfrom %}
{{indent}}) AS current_rows{%br%}
{% from table.columns as column where column.hubHashKey or column.linkHashKey or
column.keyType.name == "MultiActiveNatural" or column.keyType.name ==
"MultiActiveSequence" or column.name == "dss_tracking_flag" %}
  {% if loop.first %} ON {% else %} AND {% endif %}
  {% if column.name == "dss_tracking_flag" -%} current_rows.dss_record_source =
source_table.dss_record_source{%br%} {% else -%}{{column.name}} = current_rows.
{{column.name}}{%br%}{%- endif -%}
{% endfrom %}
{% from table.columns as column where column.name == "dss_tracking_flag"-%}
{{indent}} AND (current_rows.dss_record_source IS NOT NULL OR
stage_table.dss_record_source IS NOT NULL)
  {% endfrom -%}
{% endmacro %}

```

Multiple Source Mappings

If there is a hub or link with multiple source mappings an additional step is required after each export from 3D to Red. Select one of the source mappings for multi-source link/hub. Note the listed source table and then use **rebuild** to update the code. Select "Source" in the left window pane and then open the source join window. Update the OUTER JOIN with the correct source table and click OK to regenerate the code. Do not change the table alias. See the highlights in the example below.



- Optional - Direct metadata manipulation

WhereScape recommends performing a backup of your database before performing any database manipulation

The SQL code below runs against the Red database and will find all multi-source tracking satellite update source metadata and alter the from statement metadata to reference the source stage table. This works in concert in with the rest of the modifications suggestions and may not work as expected if the environment is different than expected. Please test the code on one or two objects before implementing. Please note that this will update the metadata, but the code for each update process will still need to be regenerated.

```

UPDATE smtab
SET smtab.smt_where = REPLACE (
    CAST(smtab.smt_where AS VARCHAR(8000))
    , REVERSE (
        SUBSTRING (
            REVERSE (CAST(smtab.smt_where AS VARCHAR(8000)))
            , (CHARINDEX (']', REVERSE (CAST(smtab.smt_where AS
VARCHAR(8000)))) + 1)
            , ((CHARINDEX ('[', REVERSE (CAST(smtab.smt_where AS
VARCHAR(8000)))) - 1
            - (CHARINDEX (']', REVERSE (CAST(smtab.smt_where AS
VARCHAR(8000))))))))))
    , smcol.smc_src_table)
FROM    dbo.ws_source_mapping_col smcol
INNER JOIN dbo.ws_normal_col col
    ON smcol.smc_parent_col_key = col.nc_col_key
INNER JOIN dbo.ws_source_mapping_tab smtab
    INNER JOIN dbo.ws_normal_tab tab
        ON smtab.smt_parent_obj_key = tab.nt_obj_key
    ON smcol.smc_source_mapping_key = smtab.smt_source_mapping_key
WHERE   tab.nt_table_name LIKE '%tracking'
    AND smcol.smc_src_column = 'dss_record_source'

```

That's a wrap!

And with that you have repeatable code for your Record Tracking Satellites. This process has proven extremely helpful to our clients and a reliable alternative when no CDC is available. We've done our best to capture all the steps we use in this process and have had success using it in MS SQL Server and Snowflake environments. If it helps you, please drop us a line and let us know! Thanks go to Wherescape for producing the base field solution that made this more robust version possible.

Please send any questions or comments to infoVia support at support@info-via.com

Change Log:

1.1.0

- Make optional `dss_tracking_flag` attribute removal steps required steps

1.0.6

- Expand model conversion rule import section

1.0.5

- Remove extra parenthesis from satellite current row modification
- Correct screenshot for multiple source mappings to highlight both needed replacements
- Add suggestion to modify a copy of model conversion rules
- Add note for snowflake syntax on `dss_load_date` changei

1.0.4 - Add versioning for Red and 3D